Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A method for controlling concurrency of access to data in a database system having a database engine, data manager and database, the method comprising:

providing for enabling and indexing of selective partition locking of a table;

partitioning ~~a~~ the table in the database system into a plurality of partitions;

receiving a lock request having one or more statements for access to data in the database system, the lock request being a request for a page lock or a row lock for a corresponding row or page in the database system containing the data;

determining a minimum lock state for each statement of the request;

identifying a partition of the plurality of partitions that contains the row or the page in the database system containing the data;

associating the lock request with a partition lock on the partition that contains the row or the page in the database system containing the data, the partition lock selectively locking the partition at ~~a~~ the minimum lock state that permits serialized access to data in the partition; and serially accessing the data in the partition using the partition lock, wherein otherwise locking by the partition lock is avoided.

2. (Previously Presented) The method of claim 1, further comprising:

responsive to the data being committed at a time of receiving the lock request, accessing the data without using the partition lock.

3. (Previously Presented) The method of claim 1, wherein the lock state protects against interference in the form of updates to the partition.

4. (Previously Presented) The method of claim 3, wherein serially accessing the data in the partition includes permitting lock requests access to the partition that are compatible with the lock state.

5. (Previously Presented) The method of claim 1, wherein serially accessing the data using the partition lock comprises an application accessing the data through a single database system.

6. (Currently Amended) The method of claim 1, wherein serially accessing the data using the partition lock comprises a second database system in a data-sharing environment accessing the data.

7-8. (Cancelled)

9. (Previously Presented) The method of claim 1, wherein the lock request is a request for a shared lock.

10. (Previously Presented) The method of claim 1, wherein the lock request is a request for an exclusive lock.

11. (Previously Presented) The method of claim 4, further comprising:

receiving a lockmax value;

accumulating for an application, a number of lock requests for access to the data in the database system by the application;

comparing the number of lock requests with the lockmax value; and

when the number of lock requests equals the lockmax value, escalating the lock state.

12. (Currently Amended) A ~~database management system implemented in a~~ digital computer system including a computer having a central processing unit and a memory unit containing computer software for operating the computer and having executable instructions for~~, the database management system configured to manage~~ managing access to data in a database system having a database engine, data manager and database, the ~~database management~~ system comprising executable instructions providing for:

providing for enabling and indexing of selective partition locking of a table;

a ~~database system component to partition a~~ partitioning the table in the ~~database~~ system into a plurality of partitions; and

~~a~~ the data manager for managing data in the system configured to:

receive a lock request having one or more statements for access to data in the database system, the lock request being a request for a page lock or a row lock for a corresponding row or page in the ~~database~~ system containing the data;

determine a minimum lock state for each statement of the request;

identify a partition of the plurality of partitions that contains the row or the page in the ~~database~~ system containing the data;

associate the lock request with a partition lock on the partition that contains the row or the page in the database system containing the data, the partition lock selectively locking the partition at ~~a~~ the minimum lock state that permits serialized access to data in the partition; and serially ~~access~~ accessing the data in the partition using the partition lock, wherein otherwise locking by the partition lock is avoided.

13. (Currently Amended) The ~~database management~~ system of claim 12, wherein the data manager is further configured to access the data without using the partition lock responsive to the data being committed at a time the data manager means received the lock request.

14. (Currently Amended) The ~~database management~~ system of claim 12, wherein the lock state protects against interference in the form of updates to the partition.

15. (Currently Amended) The ~~database management~~ system of claim 14, wherein the data manager serializes access to the partition by granting lock requests on the partition that are compatible with the lock state.

16. (Currently Amended) The ~~database management~~ system of claim 14, wherein the data manager is coupled to a plurality of database systems of a data-sharing environment.

17. (Currently Amended) The ~~database management~~ system of claim 12, wherein the lock request is from an application coupled to the ~~database management~~ system.

18. (Currently Amended) The ~~database management~~ system of claim 15, wherein the data~~base~~ manager is further configured to:

accumulate for an application a number of lock requests for access to the data in the ~~database~~ system by the application;

compare the number of lock requests with a pre-determined value; and

when the number of lock requests equals the pre-determined value, request escalation of the lock state.

19-20. (Cancelled)

21. (Currently Amended) The ~~database management~~ system of claim 12, wherein the lock request is a request for a shared lock.

22. (Currently Amended) The ~~database management~~ system of claim 12, wherein the lock request is a request for an exclusive lock.

23. (Currently Amended) A computer readable medium encoded with a computer program providing for controlling concurrency of access to data in a database system including a computer having a central processing unit and a memory unit containing computer software for operating the computer and the computer program comprising computer executable instructions for:

providing for enabling and indexing of selective partition locking of a table;

partitioning ~~a~~ the table in the database system into a plurality of partitions;

receiving a lock request having one or more statements for access to data in the database system, the lock request being a request for a page lock or a row lock for a corresponding row or page in the database system containing the data;

determining a minimum lock state for each statements of the request;

identifying a partition of the plurality of partitions that contains the row or the page in the database system containing the data;

associating the lock request with a partition lock on the partition that contains the row or the page in the database system containing the data, the partition lock selectively locking the partition at ~~a~~ the minimum lock state that permits serialized access to data in the partition; and serially accessing the data in the partition using the partition lock, wherein otherwise locking by the partition lock is avoided.

24. (Previously Presented) The computer readable medium of claim 23, wherein the computer program further comprises computer executable instructions for:

accessing the data without using the partition lock responsive to the data being committed at a time of receiving the lock request.

25. (Previously Presented) The computer readable medium of claim 23, wherein the lock state protects against interference in the form of updates to the partition.

26. (Previously Presented) The computer readable medium of claim 25, wherein the computer executable instructions for serially accessing the data in the partition include computer executable instructions for permitting lock requests access to the partition that are compatible with the lock state.

27. (Previously Presented) The computer readable medium of claim 23, wherein the computer executable instructions for serially accessing the data using the partition lock include computer executable instructions for having an application access the data through a single database system.

28. (Previously Presented) The computer readable medium of claim 23, wherein the computer executable instructions for serially accessing the data using the partition lock include computer executable instructions for having a second database system in a data-sharing environment access the data.

29-30. (Cancelled)

31. (Previously Presented) The computer readable medium of claim 23, wherein the lock request is a request for a shared lock.

32. (Previously Presented) The computer readable medium of claim 23, wherein the lock request is a request for an exclusive lock.

33. (Previously Presented) The computer readable medium of claim 26, wherein the computer program further includes computer executable instructions for:

receiving a lockmax value;

accumulating for an application, a number of lock requests for access to the data in the database system by the application;

comparing the number of lock requests with the lockmax value; and

when the number of lock requests equals the lockmax value, escalating the lock state.

34. (New) The method of claim 1, wherein the determining a minimum lock state for each statements of the request further comprises creating one or more control blocks for a called operation in relation to the one or more statements.

35. (New) The method of claim 34, further comprising copying to the one or more control blocks in relation to the table, one or more of: state of a selective partition locking, selected locking granularity, a lockmax value.

36. (New)The system of claim 12, wherein the determining a minimum lock state for each statements of the request further comprises creating one or more control blocks for a called operation in relation to the one or more statements.

37. (New) The system of claim 36, further comprising copying to the one or more control blocks in relation to the table, one or more of: state of a selective partition locking, selected locking granularity, a lockmax value.

38. (New)The computer readable medium of claim 23, wherein the determining a minimum lock state for each statements of the request further comprises creating one or more control blocks for a called operation in relation to the one or more statements.

39. (New) The medium of claim 38, further comprising copying to the one or more control blocks in relation to the table, one or more of: state of a selective partition locking, selected locking granularity, a lockmax value.